



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/716,287	11/18/2003	Sriram Devanathan	PQH03-037	6073

34225 7590 05/12/2006

UNISYS CORP.  
25725 JERONIMO ROAD, MS400  
MISSION VIEJO, CA 92691

EXAMINER
----------

SYED, FARHAN M

ART UNIT	PAPER NUMBER
----------	--------------

2165

DATE MAILED: 05/12/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

<b>Office Action Summary</b>	Application No. 10/716,287	Applicant(s) DEVANATHAN ET AL.	
	Examiner Farhan M. Syed	Art Unit 2165	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

#### Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

#### Status

- 1) ☒ Responsive to communication(s) filed on 18 November 2003.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

#### Disposition of Claims

- 4) ☒ Claim(s) 1-60 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-60 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

#### Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 18 November 2003 is/are: a) ☐ accepted or b) ☒ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

#### Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some \* c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
  2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
- \* See the attached detailed Office action for a list of the certified copies not received.

#### Attachment(s)

- |  |   |
|--|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892)  | 4) <input type="checkbox"/> Interview Summary (PTO-413)<br>Paper No(s)/Mail Date. _____ |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948)                                   | 5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152)             |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)<br>Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____  |

**DETAILED ACTION**

1. Claims 1-60 are pending.

***Drawings***

2. The drawings are objected to as failing to comply with 37 CFR 1.84(p)(5) because they include the following reference character(s) not mentioned in the description: Figure 5, block 506; Figure 9, block 916; and Figure 10, block 1010. Corrected drawing sheets in compliance with 37 CFR 1.121(d), or amendment to the specification to add the reference character(s) in the description in compliance with 37 CFR 1.121(b) are required in reply to the Office action to avoid abandonment of the application. Any amended replacement drawing sheet should include all of the figures appearing on the immediate prior version of the sheet, even if only one figure is being amended. Each drawing sheet submitted after the filing date of an application must be labeled in the top margin as either "Replacement Sheet" or "New Sheet" pursuant to 37 CFR 1.121(d). If the changes are not accepted by the examiner, the applicant will be notified and informed of any required corrective action in the next Office action. The objection to the drawings will not be held in abeyance.

3. The drawings are objected to as failing to comply with 37 CFR 1.84(p)(5) because they do not include the following reference sign(s) mentioned in the description: Figure 14, block 1404; and Figure 21, block 2114. Corrected drawing sheets in compliance with 37 CFR 1.121(d) are required in reply to the Office action to

avoid abandonment of the application. Any amended replacement drawing sheet should include all of the figures appearing on the immediate prior version of the sheet, even if only one figure is being amended. Each drawing sheet submitted after the filing date of an application must be labeled in the top margin as either "Replacement Sheet" or "New Sheet" pursuant to 37 CFR 1.121(d). If the changes are not accepted by the examiner, the applicant will be notified and informed of any required corrective action in the next Office action. The objection to the drawings will not be held in abeyance.

4. The drawings are objected to as failing to comply with 37 CFR 1.84(p)(4) because reference character "block 1408," in Figure 14, has been used to designate both "Is CWM Data-Type User-Defined?" and "Is CWM Data Type Text?." Corrected drawing sheets in compliance with 37 CFR 1.121(d) are required in reply to the Office action to avoid abandonment of the application. Any amended replacement drawing sheet should include all of the figures appearing on the immediate prior version of the sheet, even if only one figure is being amended. Each drawing sheet submitted after the filing date of an application must be labeled in the top margin as either "Replacement Sheet" or "New Sheet" pursuant to 37 CFR 1.121(d). If the changes are not accepted by the examiner, the applicant will be notified and informed of any required corrective action in the next Office action. The objection to the drawings will not be held in abeyance.

### ***Specification***

5. 35 U.S.C. 112, first paragraph, requires the specification to be written in "full, clear, concise, and exact terms." The specification is replete with terms that are not clear, concise and exact. The specification should be revised carefully in order to comply with 35 U.S.C. 112, first paragraph. Examples of some unclear, inexact or verbose terms used in the specification are: design items, design libraries, design model, subject areas, and generalization.

### ***Double Patenting***

6. The nonstatutory double patenting rejection is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the "right to exclude" granted by a patent and to prevent possible harassment by multiple assignees. See *In re Goodman*, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); *In re Longi*, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); *In re Van Ornum*, 686 F.2d 937, 214 USPQ 761 (CCPA 1982); *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970); and, *In re Thorington*, 418 F.2d 528, 163 USPQ 644 (CCPA 1969).

A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) may be used to overcome an actual or provisional rejection based on a nonstatutory double patenting ground provided the conflicting application or patent is shown to be commonly owned with this application. See 37 CFR 1.130(b).

Effective January 1, 1994, a registered attorney or agent of record may sign a terminal disclaimer. A terminal disclaimer signed by the assignee must fully comply with 37 CFR 3.73(b).

Claims 1-3, 21-23, and 41-43 provisionally rejected under the judicially created doctrine of obviousness-type double patenting as being unpatentable over claims 1-3,

Art Unit: 2165

13-17, and 27-29 of copending Application No. 10/716,286. Although the conflicting claims are not identical, they are not patentable distinct from each other because they are substantially similar in scope and they use the same limitations.

1-3, 13-17, and 27-29 of 10/716,286 reference recites all the elements of claims 1-3, 21-23, and 41-43 of the instant application 10/716,287 and as such anticipates claims 1-3, 21-23, and 41-43 of the instant application.

“A later patent claim is not patentably distinct from an earlier patent claim if the later claim is obvious over, or anticipated by, the earlier claim. In re Longi, 759 F.2d at 896, 225 USPQ at 651 (affirming a holding of obviousness-type double patents); In re Berg, 140 F.3d at 1437, 46 USPQ2d at 1233 (Fed. Cir. 1998) (affirming a holding of obviousness-type double patenting where a patent application claim to a genus is anticipated by a patent claim to a species within that genus).” ELI LILLY AND COMPANY v. BARR LABORATORIES, INC., United States Court of Appeals for the Federal Circuit, ON PETITION FOR REHEARING EN BANC (DECIDED: May 30, 2001).

It would have been obvious to a person of ordinary skill in the art at the time of the invention was made to omit the additional elements of claims 1-3, 13-17, and 27-29 to arrive at the 1-3, 21-23, and 41-43 of the instant application because the person would have realized that the remaining element would perform the same functions as before. “Omissions of element and its function in combination is obvious expedient if the remaining elements perform same functions as before.” See In re Karlson (CCPA) 136

USPQ 184, decide Jan 16, 1963, App. No. 6857, U. S. Court of Customs and Patent Appeals.

This is a provisional obviousness-type double patenting rejection because the conflicting claims have not in fact been patented.

### ***Claim Rejections - 35 USC § 101***

7. 35 U.S.C. 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

Claims 21-40 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

As per claims 21-40, the claim clearly recite a "An article of manufacture comprising of: a machine-accessible medium including data that, when accessed by a machine, causes the machine to perform the operation of:" The Applicant uses the word "machine-accessible medium" on page 10 of the specification, which recites "the program or code segments can be stored in a process or a machine accessible medium or transmitted by a computer data signal embodied in a carrier wave, or a signal modulated by a carrier, over a transmission medium. Examples of the processor readable or machine accessible medium include an electronic circuit, a semiconductor memory device, a read only memory (ROM), a flash memory, an erasable ROM (EROM), a floppy diskette, a compact disk (CD) ROM, an optical disk, a hard disk, a fiber optic medium, a radio frequency (RF) link, etc. The computer data signal may

Art Unit: 2165

include any signal that can propagate over a transmission medium such as electronic network channels, optical fiber, air, electromagnetic, RF links, etc.” Based on these references, the Examiner understands that implementing the claim mentioned above would render the result of the claim as intangible. A signal-bearing medium is not tangible, and cannot tangibly embody a computer program or process since a computer cannot understand/realize (i.e. execute) the computer program or process when embodied on the data signal. Computer program or processes are only realized within the computer when stored in a memory or storage element (such as RAM or ROM). Therefore, a data signal does not meet the “useful, concrete, and tangible” requirement as set forth in *State Street*, 149 F.3d at 1373, 47 USPQ2d at 1601-02, and hence claims 25-32 are non statutory under 35 U.S.C. 101.

### ***Claim Rejections - 35 USC § 102***

1. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

2. Claims 1-60 are rejected under 35 U.S.C. 102(b) as being anticipated by a non-patent literature titled "Designing and Creating Relational Schemas with a CWM-Based Tool" by Kumpon Farpinyo and Twittie Senivongse, pages 456-461, 2002 (known hereinafter as Farpinyo).



As per claims 1, 21, and 41, Farpinyo teaches a method comprising: converting physical aspects of a common warehouse model (CWM) to corresponding database management system (DBMS) items in a relational database, by processing in a hierarchical manner the physical aspects and creating the corresponding DBMS items, the physical aspects comprising relational catalogs, the relational catalogs comprising relational schemas, the corresponding DBMS items comprising DBMS catalogs, the DBMS catalogs comprising DBMS schemas (i.e. *"This paper presents a design and development of a tool called ER2CWM that creates CWM relational database schemas from physical data models represented by ER diagrams. The tool supports the creation of ER diagrams, transformation into CWM format, and creation of database schemas for relational database management systems. It can also transform database schemas back into CWM and ER diagrams respectively."* "ER diagrams are generally used to express designs of relational databases [1]. There are tools, such as PowerDesigner [2] and Erwin [3], that can help database designers to design a database with ER diagrams and create database schemas. These tools usually support the reverse of the process to create ER diagrams from existing database schemas also. All these are done via intermediate schema representations that are specific to individual design tools. This means, for example, PowerDesigner and Erwin both have their own metadata format that represents ER models and is used to create database schemas. This situation is not convenient for the designers to export a database schema designed and created by one tool to other working environments since specific mapping between the metadata of the source environment and the one understood by the target will be required for each pair of the exchanging environment.") (Abstract; page 456, paragraph 1).

As per claims 2, 22, and 42, Farpinyo teaches a method wherein converting comprises the operations of: (a) scanning through the relational catalogs (i.e. "DBMS

*Information – This module, via JDBC, creates database schema from CWM Relational metadata, reads in existing database schemas to create CWM Relational metadata and ER diagrams, and maintains information about DBMSes that ER2CWM supports, i.e. SQL data types and database commands for creating and reading in schemas. DBMSes on which ER2CWM have been tested are Sybase Adaptive Server v.11.9.2 [9] and Microsoft SQL Server 2000 [10]. Other DBMSes can be supported by providing ER2CWM with .jar files that contain corresponding DBMS information.*” The preceding text clearly indicates that ER2CWM supports SQL data types and database commands for creating and reading schemas. It is clear that in order to create and read schemas, scanning of ER libraries must be performed. Figure 3 clearly illustrates such example.)(Page 459, paragraph 3); (b) for a first of the relational catalogs, creating a corresponding first DBMS catalog in the relational database; (c) for each of the relational schemas in the first relational catalog, creating a corresponding DBMS schema in the corresponding DBMS catalog to hold corresponding information; and (d) processing each of the relational schemas to produce corresponding information for the corresponding DBMS schema (For the remainder of steps b-d are steps found in the user manual of the ER2CWM tool.)(Page 461, paragraph 1).

As per claims 3, 23, and 43, Farpinyo teaches a method wherein, in operation (d), each of the relational schemas is processed independently (Figures 1-11 steps through the process of creating an ER model and then converting it into a relational database. This process is a continuous process, where each ER model is created independently from the other, until the user completes the desired relational database specifications.)(Figures 1-11).

As per claims 4, 24, and 44, Farpinyo teaches a method wherein operation (d) comprises: (1) processing CWM data types included in a first of the relational schemas (i.e. *"Also, in Figure 1, database design tools can be used to design database schemas in a usual way. But to incorporate CWM for easy exchange, there specific metadata representations have to be transformed to CWM format using a tool called Meta Integration Model Bridge (MIMB)."* *"This paper presents a design and development of a database design tool, called ER2CWM that can be used to design relational databases and also create schemas for particular DBMSes by using CWM as its metadata format."* The preceding text clearly indicates that CWM data types, which are contained in CWM metadata format are processed for the first of the relational schemas, which are schemas for particular DBMSes. )(Page 457, paragraph 1); (2) creating DBMS data types corresponding to the CWM data types (i.e. *"This paper presents a design and development of a database design tool, called ER2CWM that can be used to design relational databases and also create schemas for particular DBMSes by using CWM as its metadata format."* The preceding text clearly indicates that creating DBMS data types corresponding to CWM data types occurs in the process of designing a relational database to create schemas for DBMSes using CWM as its metadata format. An ordinary person skilled in the art would understand that such a step is required.)(Page 457, paragraph 1); (3) processing relational tables included in the first relational schema (i.e. *"This paper presents a design and development of a database design tool, called ER2CWM that can be used to design relational databases and also create schemas for particular DBMSes by using CWM as its metadata format."* The preceding text clearly indicates that processing relational tables is contained in the design of a relational database and when creating schemas, an ordinary person skilled in the art would find a first relational schema associated with a relational table.)(Page 457, paragraph 1); (4) processing relational foreign key relationships for each of the relational tables (i.e. *"ER2CWM supports CWM specification version 1.0. Its relational part, CWM Relational, consists of several elements including the following: ForeignKey refers to feature which references the XMI ID of the column that is the foreign key of the table. It likes to the*

*primary key of another table by a uniquekey.”*(Page 458); (5) processing relational checkconstraints for the first relational schema (i.e. *“ER2CWM supports CWM specification version 1.0. Its relational part, CWM Relational, consists of several elements including the following: CheckConstraint refers to the constraint for the value of a particular column.”*)(Page 458); (6) creating DBMS tables corresponding to the relational tables (i.e. *“DBMS Information – This module, via JDBC, creates database schemas from CWM Relational metadata, reads in existing database schemas to create CWM Relational metadata and ER diagrams, and maintains information about DBMSes that ER2CWM supports.”*)(Page 459); (7) processing relational views for the first relational schema (i.e. *“ER Module – This connects together the three modules above. It contacts DBMS Information module when database designers select DBMSes to design physical data models or to create database schemas. It interacts with the Metadata module to get and save DIA and CWM Relational metadata.”* The preceding text clearly indicates that the processing of relational views for the schema occurs in the ER module.)(Page 459); (8) processing relational indices for the first relational schema; (9) processing relational triggers for the first relational schema; and (10) processing relational procedures for the first relational schema (i.e. *“ER Module – This connects together the three modules above. It contacts DBMS Information module when database designers select DBMSes to design physical data models or to create database schemas. It interacts with the Metadata module to get and save DIA and CWM Relational metadata.”* An ordinary person skilled in the art understands that steps 8-10 would be performed within the ER Module.)(Page 459).

As per claims 5, 25, and 45, Farpinyo teaches a method wherein (1) processing CWM data types included in a first of the relational schemas comprises: for one of the CWM data types, determining whether the CWM data type is user-defined; if the CWM data type is user-defined, obtaining base type and constraint of the CWM data type; and

if the CWM data type is text, obtaining a character set, name of language and collation sets associated with the CWM data type (i.e. *"Also, in Figure 1, database design tools can be used to design database schemas in a usual way. But to incorporate CWM for easy exchange, there specific metadata representations have to be transformed to CWM format using a tool called Meta Integration Model Bridge (MIMB)."* *"This paper presents a design and development of a database design tool, called ER2CWM that can be used to design relational databases and also create schemas for particular DBMSes by using CWM as its metadata format."* The preceding text clearly indicates that CWM data types, which are contained in CWM metadata format are processed for the first of the relational schemas, which are schemas for particular DBMSes and that determining whether the CWM data type is user-defined, obtaining base type and constraint of CWM data type, and obtaining a character set, name of language and collation set associated with the CWM data type is contained within using Meta Integration Model Bridge.)(Page 457, paragraph 1).

As per claims 6, 26, and 46, Farpinyo teaches a method wherein (2) creating DBMS data types corresponding to the CWM data types comprises: for a first of the CWM data types that is user-defined, creating a corresponding DBMS data type in the corresponding DBMS schema; setting physical type for the DBMS data type, based on the obtained base type of the first CWM data type; and binding a constraint to the DBMS data type, based on the obtained constraint of the first CWM data type (i.e. *"This paper presents a design and development of a database design tool, called ER2CWM that can be used to design relational databases and also create schemas for particular DBMSes by using CWM as its metadata format."* The preceding text clearly indicates that creating DBMS data types corresponding to CWM data types occurs in the process of designing a relational database to create schemas for DBMSes using CWM as its metadata format. An ordinary person skilled in the art would understand that such a step is required. Furthermore the steps described in this claim to enhance DBMS data types to

correspond to CWM data type are consistent within using the ER2CWM to design a relational database.)(Page 457, paragraph 1).

As per claims 7, 27, and 47, Farpinyo teaches a method wherein (3) processing relational tables included in the first relational schema comprises: determining whether there is a first relational table in the first relational schema (i.e. *"This paper presents a design and development of a database design tool, called ER2CWM that can be used to design relational databases and also create schemas for particular DBMSes by using CWM as its metadata format."* The preceding text clearly indicates that processing relational tables is contained in the design of a relational database and when creating schemas, an ordinary person skilled in the art would find a first relational schema associated with a relational table.)(Page 457, paragraph 1); if there is a first relational table in the first relational schema, then: determining relational relational table having a relational columns: obtaining column properties including type, precision, scale, length, IsNullable, CollationName, and CharactersetName (i.e. *"Column refers to a column name within a table. It references XMI ID of a SQL Datatype and the table that owns it, e.g. <CWMRDB:Column xmi.id="\_12" name="sex" isNullable="columnNullable" visibility="public" length="1" type="\_13" owner="\_3"></CWMRDB:Column>"*)(Page 458); columns in the first relational table, the first relational primary key (i.e. *"PrimaryKey refers to feature which references the XMI ID of the column that is the primary key of the table, e.g. <CWMRDB:PrimaryKey xmi.id="\_16" name="PK\_STUDENT" visibility="public" namespace="\_3" feature="\_4"/>"*)(Page 458); and, for each of the verifying that the obtained type types; matches one of the DBMS data determining whether the relational column is pad of the relational primary key; and flagging the relational column if the relational column is part of the relational primary key (i.e. *"DBMS Information – This module, via JDBC, creates database schemas from CWM Relational metadata, reads in existing*

Art Unit: 2165

*database schemas to create CWM Relational metadata and ER diagrams, and maintains information about DBMSes that ER2CWM supports, i.e. SQL data types and database commands for creating and reading in schemas. DBMSes on which ER2CWM have been tested are Sybase Adaptive Server v.11.9.2 [9] and Microsoft SQL Server 2000 [10]. Other DBMSes can be supported by providing ER2CWM with .jar files that contain corresponding DBMS information.”(Page 459).*

As per claims 8, 28, and 48, Farpinyo teaches a method wherein (4) processing relational foreign key relationships for each of the relational tables comprises (i.e. “ER2CWM supports CWM specification version 1.0. Its relational part, CWM Relational, consists of several elements including the following: ForeignKey refers to feature which references the XMI ID of the column that is the foreign key of the table. It links to the primary key of another table by a uniquekey.”)(Page 458): for a first of the relational tables, enumerating child relational tables having foreign key relationships with the first relational table (i.e. “ForeignKey refers to feature which references the XMI ID of the column that is the foreign key of the table. It links to the primary key of another table by uniquekey, e.g. <CWMRDB:ForeignKey xmi.id=“\_17” name=“school\_id” visibility=“public” namespace=“\_3” feature=“\_10” uniqueKey=“\_18” deleteRule=“importedKeyRestrict” updateRule=“importedKeyRestrict” />”)(Page 458); for each of the foreign key relationships, determining relational columns imported from the respective child relational table to the first relational table (i.e. “Column refers to a column name within a table. It references XMI ID of a SQL Datatype and the table that owns it, e.g. <CWMRDB:Column xmi.id=“\_12” name=“sex” isNullable=“columnNullable” visibility=“public” length=“1” type=“\_13” owner=“\_3”></CWMRDB:Column>”)(Page 458); and obtaining properties of each of the imported relational columns, including “update” and “delete” referential integrity rules and deferability type (i.e. “ER Editor – This is the editor for designing physical data models with ER diagrams based on CODASYL [8]. It is also a GUI of ER2CWM; database designers can create CWM

*Relational metadata, select DBMSes to create database schemas, or create CWM Relational metadata and ER diagrams from existing relational databases.”)(Page 458).*

As per claims 9, 29, and 49, Farpinyo teaches a method wherein (5) processing relational checkconstraints for the first relational schema comprises (i.e. “ER2CWM supports CWM specification version 1.0. Its relational part, CWM Relational, consists of several elements including the following: CheckConstraint refers to the constraint for the value of a particular column.”)(Page 458): determining relational checkconstraints associated with the first relational schema; obtaining parameters associated with a first of the relational checkconstraints; and enumerating relational columns having references to the first relational checkconstraint (i.e. “ER Editor – This is the editor for designing physical data models with ER diagrams based on CODASYL [8]. It is also a GUI of ER2CWM; database designers can create CWM Relational metadata, select DBMSes to create database schemas, or create CWM Relational metadata and ER diagrams from existing relational databases.” The preceding text clearly indicates that the ER Editor would contain the limitations included in this claim to process the checkconstraint for the relational schema.)(Page 458).

As per claims 10, 30, and 50, Farpinyo teaches a method wherein (6) creating DBMS tables corresponding to the relational tables comprises: selecting from the relational tables included in the first relational schema first tables having no dependencies on any other of the relational tables; and creating a corresponding DBMS table for each of the first selected tables (i.e. “DBMS Information – This module, via JDBC, creates database schemas from CWM Relational metadata, reads in existing database schemas to



*create CWM Relational metadata and ER diagrams, and maintains information about DBMSes that ER2CWM supports.”(Page 459).*

As per claims 11, 31, and 51, Farpinyo teaches a method further comprising:  
selecting from the relational tables included in the first relational schema a second table having dependency on at least one of the first selected tables; and creating a corresponding DBMS table for the second selected table (i.e. *“DBMS Information – This module, via JDBC, creates database schemas from CWM Relational metadata, reads in existing database schemas to create CWM Relational metadata and ER diagrams, and maintains information about DBMSes that ER2CWM supports.”* The limitations stated in this claim are contained with the DBMS information that creates database schemas and tables for to correspond between the CWM information and DBMS table.)(Page 459).

As per claims 12, 32, and 52, Farpinyo teaches a method further comprising:  
selecting from the relational tables included in the first relational schema a third table having dependency on at least one of the second and the first selected tables; and creating a corresponding DBMS table for the third selected table (i.e. *“DBMS Information – This module, via JDBC, creates database schemas from CWM Relational metadata, reads in existing database schemas to create CWM Relational metadata and ER diagrams, and maintains information about DBMSes that ER2CWM supports.”* The limitations stated in this claim are contained with the DBMS information that contains dependencies between the tables, as an ordinary person skilled in the art would understand that dependencies exist between multiple databases contained in a DBMS.)(Page 459).

As per claims 13, 33, and 53, Farpinyo teaches a method further comprising: creating a corresponding DBMS table for each of mutually dependent tables from the relational tables using forward references or ALTER TABLE commands (i.e. "DBMS Information – This module, via JDBC, creates database schemas from CWM Relational metadata, reads in existing database schemas to create CWM Relational metadata and ER diagrams, and maintains information about DBMSes that ER2CWM supports." The limitations stated in this claim are contained with the DBMS information.)(Page 459).

As per claims 14, 34, and 54, Farpinyo teaches a method wherein creating a corresponding DBMS table comprises: creating DBMS columns corresponding to columns of the corresponding relational table; setting properties including precision, scale, IsNullable, CollationName, and CharactersetName for length, data type, each of the DBMS columns based on respective properties of the corresponding relational column (i.e. "Column refers to a column name within a table. It references XMI ID of a SQL Datatype and the table that owns it, e.g. <CWMRDB:Column xmi.id="\_12" name="sex" isNullable="columnNullable" visibility="public" length="1" type="\_13" owner="\_3" > </CWMRDB:Column>")(Page 458); if one of the DBMS columns is the only one of the DBMS columns that represents a primary key or a foreign key, adding property of primary key or foreign key to the one DBMS column (i.e. "PrimaryKey refers to feature which references the XMI ID of the column that is the primary key of the table, e.g. <CWMRDB:PrimaryKey xmi.id="\_16" name="PK\_STUDENT" visibility="public" namespace="\_3" feature="\_4"/> ForeignKey refers to feature which references the XMI ID of the column that is the foreign key of the table. It links to the primary key of another table by uniquekey, e.g. <CWMRDB:ForeignKey xmi.id="\_17" name="school\_id" visibility="public" namespace="\_3" feature="\_10" uniqueKey="\_18" deleteRule="importedKeyRestrict" updateRule="importedKeyRestrict" />)(page 458);

and if there is a checkconstraint associated with one of the DBMS columns and not involving any of the remaining DBMS columns, specifying the checkconstraint as column-level constraint (i.e. "CheckConstraint refers to the constraint for the value of a particular column, e.g. <CWMRDB:CheckConstraint xmi.id="\_18" name="CK\_student\_sex" visibility="public"constrainedElement="\_17" namespace="\_2"> <CWM:Constraint.body>  
<CWM:BooleanExpression xmi.id="\_19" language=""> <CWM:Expression.body>([sex] = 'M')</CWM:Expression.body> </CWM:BooleanExpression> </CWM:Constraint.body> </CWMRDB:CheckConstraint>") (Page 458).

As per claims 15, 35, and 55, Farpinyo teaches a method further comprising: if there is a multi-column primary key or a multi-column foreign key in the relational table, specifying the multi-column primary key or a multi-column foreign key in the DBMS table at table-level and identifying the DBMS columns that represent the multi-column primary key or a multi-column foreign key (i.e. "PrimaryKey refers to feature which references the XMI ID of the column that is the primary key of the table, e.g. <CWMRDB:PrimaryKey xmi.id="\_16" name="PK\_STUDENT" visibility="public" namespace="\_3" feature="\_4"/> ForeignKey refers to feature which references the XMI ID of the column that is the foreign key of the table. It links to the primary key of another table by uniquekey, e.g. <CWMRDB:ForeignKey xmi.id="\_17" name="school\_id" visibility="public" namespace="\_3" feature="\_10" uniqueKey="\_18" deleteRule="importedKeyRestrict" updateRule="importedKeyRestrict" />)(page 458); and if there is a checkconstraint involving multiple DBMS 'columns, specifying the constraint in the DBMS table at table-level and identifying the involved DBMS columns (i.e. "CheckConstraint refers to the constraint for the value of a particular column, e.g. <CWMRDB:CheckConstraint xmi.id="\_18" name="CK\_student\_sex" visibility="public"constrainedElement="\_17" namespace="\_2"> <CWM:Constraint.body>

`<CWM:BooleanExpression xmi.id="_19" language=""> <CWM:Expression.body>([sex] = 'M')</CWM:Expression.body> </CWM:BooleanExpression> </CWM:Constraint.body> </CWMRDB:CheckConstraint>") (Page 458).`

As per claims 16, 36, and 56, Farpinyo teaches a method further comprising: specifying a foreign key in the DBMS table, including: identifying a child DBMS table and DBMS columns being imported from the child DBMS table; and specifying properties of the foreign key, the properties including "update" and "delete" referential integrity rules and deferability type (i.e. *"ER Editor – This is the editor for designing physical data models with ER diagrams based on CODASYL [8]. It is also a GUI of ER2CWM; database designers can create CWM Relational metadata, select DBMSes to create database schemas, or create CWM Relational metadata and ER diagrams from existing relational databases. Metadata – This module creates and maintains two types of metadata. Diagram metadata (DIA) is the metadata of ER models with display information for the diagrams. CWM metadata represents ER models or database schemas and conforms to CWM v.1.0. DBMS Information – This module, via JDBC, creates database schemas from CWM Relational metadata, reads in existing database schemas to create CWM Relational metadata and ER diagrams, and maintains information about DBMSes that ER2CWM supports, i.e. SQL data types and database commands for creating and reading in schemas. DBMSes on which ER2CWM have been tested are Sybase Adaptive Server v.11.9.2 [9] and Microsoft SQL Server 2000 [10]. Other DBMSes can be supported by providing ER2CWM with .jar files that contain corresponding DBMS information. ER Module – This connects together the three modules above. It contacts DBMS Information module when database designers select DBMSes to design physical data models or to create database schemas. It interacts with the Metadata module to get and save DIA and CWM Relational metadata. It also controls printing of ER models as printer or HTML files."* The limitations contained with this claim are addressed based on the combination of a user using the ER Module.)(Page 458).

As per claims 17, 37, and 57, Farpinyo teaches a method wherein (7) processing relational views for the first relational schema comprises: determining relational views associated with the first relational schema; for each of the relational views: creating a corresponding DBMS view; specifying updatability of the corresponding DBMS view; and specifying query expression defining the corresponding DBMS view (i.e. *"ER Module – This connects together the three modules above. It contacts DBMS Information module when database designers select DBMSes to design physical data models or to create database schemas. It interacts with the Metadata module to get and save DIA and CWM Relational metadata."*) The preceding text clearly indicates that the processing of relational views for the schema occurs in the ER module. In addition, the limitations contained within this claim are contained in the ER Module, as these are steps that an ordinary person skilled in the art would take to process the relational views.)(Page 459).

As per claims 18, 38, and 58, Farpinyo teaches a method wherein (8) processing relational indices for the first relational schema comprises: determining relational indices associated with a first of the relational schemas; for each of the relational indices: creating a corresponding DBMS index to represent the relational index; specifying DBMS columns used by the corresponding DBMS index; and setting properties of the specified DBMS columns including IsNullable, Filtercondition, and Autoupdate (i.e. *"ER Module – This connects together the three modules above. It contacts DBMS Information module when database designers select DBMSes to design physical data models or to create database schemas. It interacts with the Metadata module to get and save DIA and CWM Relational metadata."*) The limitations contained within this claim are contained in the ER Module, as these are steps that an ordinary person skilled in the art would take to process the relational views.)(Page 459).

As per claims 19, 39, and 59, Farpinyo teaches a method wherein (9) processing relational triggers for the first relational schema comprises: determining relational triggers associated with the first relational schema; for each of the relational triggers: creating a corresponding DBMS trigger setting properties of the corresponding DBMS trigger based on properties of the relational trigger, the relational trigger monitoring a relational table; and setting a monitored DBMS table corresponding to the monitored relational table (i.e. *"ER Module – This connects together the three modules above. It contacts DBMS Information module when database designers select DBMSes to design physical data models or to create database schemas. It interacts with the Metadata module to get and save DIA and CWM Relational metadata."* The limitations contained within this claim are contained in the ER Module, as these are steps that an ordinary person skilled in the art would take to process the relational views.)(Page 459).

As per claims 20, 40, and 60, Farpinyo teaches a method wherein (10) processing relational procedures for the first relational schema comprises: determining relational procedures associated with the first relational schema; for each of the relational procedures: creating a corresponding DBMS procedure; and setting arguments for the corresponding DBMS procedure based on arguments of the relational procedure (i.e. *"ER Module – This connects together the three modules above. It contacts DBMS Information module when database designers select DBMSes to design physical data models or to create database schemas. It interacts with the Metadata module to get and save DIA and CWM Relational metadata."* The limitations contained within this claim are contained in the ER Module, as these are steps that an ordinary person skilled in the art would take to process the relational views.)(Page 459).


***Contact Information***

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Farhan M. Syed whose telephone number is 571-272-7191. The examiner can normally be reached on 8:30AM-5:00 PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Jeffrey Gaffin can be reached on 571-272-4146. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

FMS

  
JEFFREY GAFFIN  
SUPERVISORY PATENT EXAMINER  
TECHNOLOGY CENTER 2100